

# 基于Raft的MySQL高可用组件xenon实战

知数堂 吴炳锡





# 吴炳锡

知数堂联合创始人

3306π社区联合创始人

---

现在知数堂主要担任 MySQL实战班讲师，前新媒传信首席DBA，ChinaUnix.net MySQL版版主；担任多年DTCC，SACC，Oracle技术嘉年华讲师及顾问；10年+ MySQL从业经验，有多年MySQL系统架构设计及培训教学经验。擅长MySQL大规模运维管理优化、高可用方案、多IDC架构设计，企业级数据库设计等。



# 大纲

- 本次培训目标
- xenon的介级及特点
- 业界高可用实现区别对比
- xenon的raft实现
- xenon部署及使用
- 对xenon使用上的一些想法

# 本次培训目标

- 起因
  - MySQL现在主动的结构： GTID复制，增强半同步， MGR等。 原来的高可用方案，有点out了。 xenon是基于GTID+增强半同步一个高可用方案
  - xenon整体结构比较轻，通过学习xenon方便在上面做二次开发
- 目标
  - 学会xenon的部署
  - 了解xenon的程序结构

# xenon介绍及特点

- xenon是青云RadonDB底层的MySQL高可用实现
  - 开源网址: <https://github.com/radondb/xenon>
  - xenon的限制
    - 基于MySQL GTID+Row的GTID实现
    - 基于MySQL增强半同步要求
- 优点: 面向金融环境, 对于数据复制更安全

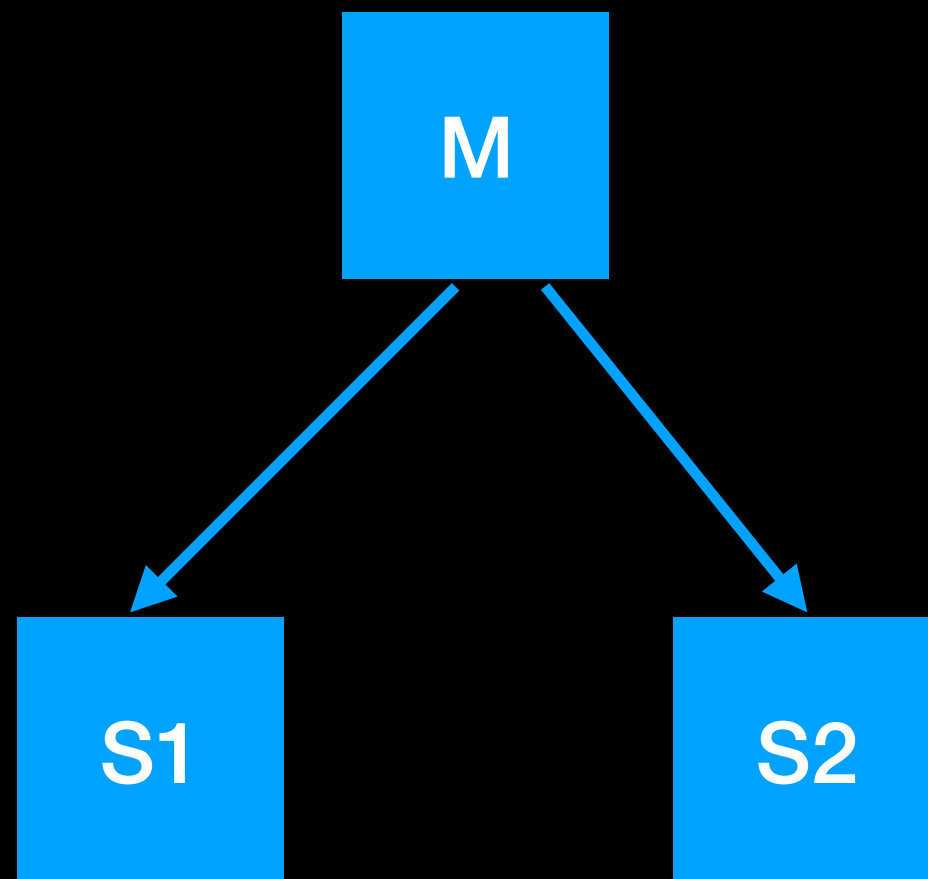
# 业界高可用实现区别对比

- 流行高可用软件：
  - MHA（MySQL GTID出现前基本是行业老大）
  - MySQL repplication mananger
  - orchestrator
- 新架构：
  - PXC
  - MGR

# 业界高可用实现区别对比

- 对于MySQL的复制增强
  - Binlog server(mysqlbinlog mysql-ripple)
  - 增强半同步
  - GTID的auto-position特性

# 业界高可用实现区别对比



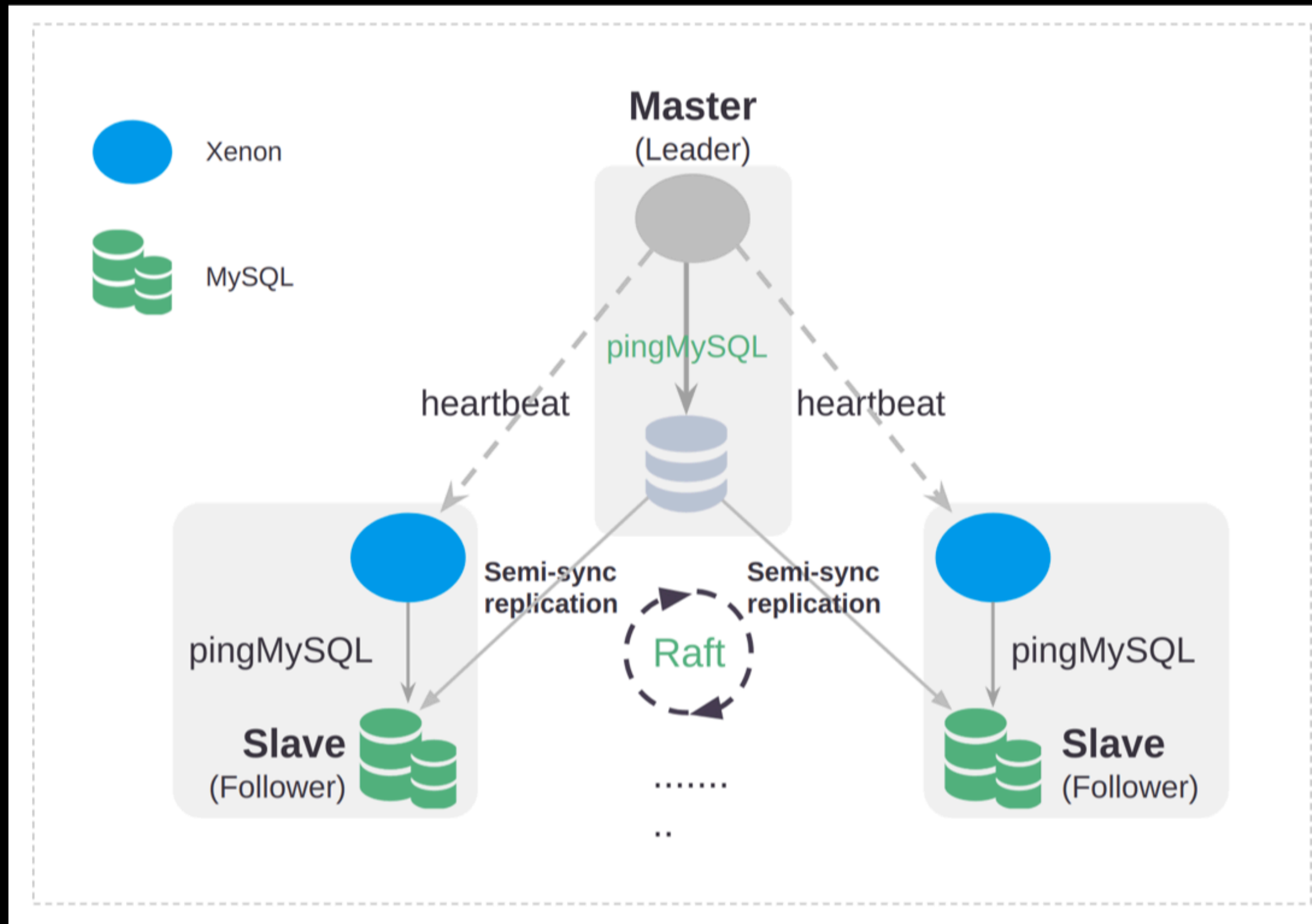
假设在GTID+Row格式下

1. 半同步是不是绝对靠谱?
2. 假设M故障, S1同步数据比S2, S2是不是能获取到多的数据?
3. 增强半同步是不是就不会丢数据?



# xenon的raft实现

- xenon架构



1. 每个节点一个Agent
2. Agent之间利用Raft先举Leader
3. Leader和Master在一起，同时持有VIP

# xenon部署及使用

1. Linux环境准备
2. golang编译环境
3. 源码下载 & xenon编译 & MySQL环境部署
4. xenon配置
5. xenon基于VIP部署
6. xenon高可用切换实现
7. xenon故障切换

# Linux环境准备

基于安装节点:

机器名	ip	安装角色
dzst141	172.18.0.10	xenon,mysql
dzst142	172.18.0.11	xenon,mysql
dzst143	172.18.0.12	xenon,mysql

- 主机间ssh信任建立 (172.18.0.10)
  - ssh-keygen
  - cd .ssh/
  - cat id\_rsa.pub >>authorized\_keys
  - chmod 600 id\_rsa
  - chmod 600 authorized\_keys
- #scp -r .ssh 172.18.0.11:~/
- #scp -r .ssh 172.18.0.12:~/

# golang编译环境

1. `wget https://dl.google.com/go/go1.11.2.linux-amd64.tar.gz`

2. `tar -xvf go1.11.2.linux-amd64.tar.gz -C /usr/local`

3. `echo "export PATH=$PATH:/usr/local/go/bin" >>/etc/profile`

4. `source /etc/profile`

- 验证Go安装是否成功

```
# go version
go version go1.11.2 linux/amd64
```

# 源码下载&部署

1.git clone git@github.com:radondb/xenon.git

2.cd xenon

3.make

- 确认： bin目录下生成了xenon, xenoncli
- 部署
  - # mkdir /data/xenon
  - # cp -r bin /data/xenon/
  - # mkdir /etc/xenon
  - # cp xenon.json /etc/xenon/
  - # echo "/etc/xenon/xenon.json" >/data/xenon/bin/config.path

基于安装节点：

机器名	ip	安装角色
dzst141	172.18.0.10	xenon,mysql
dzst142	172.18.0.11	xenon,mysql
dzst143	172.18.0.12	xenon,mysql

**#scp -r /data/xenon 172.18.0.11:/data/**

**#scp -r /data/xenon 172.18.0.12:/data/**



# MySQL环境部署

## 1. 基本准备

### ◦ 下载mysql

```
cd /data  
wget https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.22-linux-glibc2.12-x86_64.tar.gz
```

### ◦ MySQL软件放置：

```
mkdir /opt/mysql  
cd /opt/mysql/  
tar zxvf /data/mysql/mysql-5.7.22-linux-glibc2.12-x86_64.tar.gz
```

### ◦ basedir: /usr/local/mysql

```
ln -s /opt/mysql/mysql-xxx /usr/local/mysql
```

## 2. 创建数据库相关目录：

```
mkdir /data/mysql/ -p  
mkdir /data/mysql/mysql3306/{data,logs,tmp} -p
```

## 3. 创建mysql用户

```
groupadd mysql  
useradd -g mysql -s /sbin/nologin -d /usr/local/mysql -MN mysql
```

# MySQL环境部署

## 4. 更改权限

```
chown -R mysql:mysql /data/mysql/mysql3306  
chown -R mysql:mysql /opt/mysql/  
chown -R mysql:mysql /usr/local/mysql
```

## 5. 配置文参考 /data/mysql/mysql3306/my3306.cnf 参考URL

## 6. 初始化

```
/usr/local/mysql/bin/mysqld --defaults-file=/data/mysql/mysql3306/my3306.cnf --initialize
```

确认没有错误提示，从error log中找到初始化的密码。

```
grep "password" /data/mysql/mysql3306/data/error.log
```

## 7. 第一次启动MySQL

```
/usr/local/mysql/bin/mysqld --defaults-file=/data/mysql/mysql3306/my3306.cnf &
```

# MySQL环境部署

- MySQL备份及监控用的帐号
  - `mysql>create user 'xenon'@'127.0.0.1';`
  - `mysql>grant all privileges on *.* to 'xenon'@'127.0.0.1';`

# MySQL环境部署

- 加载半同步plugin
  - `mysql>INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';`
  - `mysql>INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';`
- 清理MySQL的日志：
  - `reset master;`
- 安装xtrabackup
  - `yum localinstall -y percona-xtrabackup-24-2.4.12-1.el7.x86_64.rpm`

# MySQL环境部署

- MySQL安装注意事项
  - 可以一台机器上安装好，让这个节点成为Leader
  - 其它节点可以利用xenon的rebuildme功能拉起来



# xenon配置

- <https://github.com/zhishutech/radon-ebook/blob/master/xenon/3-xenon-conf.md>
- `cat /etc/xenon/xenon.json`

```
{
  "server":
  {
    "节点": "本机IP",
    "endpoint": "172.18.0.10:8801"
  },

  "raft":
  {
    "meta-datadir": "raft.meta",
    "heartbeat-timeout": 2000,
    "election-timeout": 6000,
    "Raft中成为leader后要执行系统命令": "这个地方也可以支持脚本调用",
    "leader-start-command": "ip a a 172.18.0.100/16 dev eth0 && arping -c 3 -A 172.18.0.100 -I eth0",
    "Raft从Leader切换为follower时执行的命令": "这个地方也可以支持脚本调用",
    "leader-stop-command": "ip a d 172.18.0.100/16 dev eth0"
  },
}
```

# xenon配置

```
"mysql":  
{  
  "admin": "root",  
  "passwd": "",  
  "host": "127.0.0.1",  
  "port": 3306,  
  "basedir": "/usr/local/mysql",  
  "MySQL启动的配置文件": "",  
  "defaults-file": "/data/mysql/mysql3306/my3306.cnf",  
  "切换为master时执行的命令": "",  
  "master-sysvars": "super_read_only=0,read_only=0,innodb_flush_log_at_trx_commit=1,sync_binlog=1",  
  "切换为slave时执行的命令": "",  
  "slave-sysvars": "super_read_only=1,read_only=1,innodb_flush_log_at_trx_commit=2,sync_binlog=0"  
},
```

# xenon配置

```
"replication":
{
  "复制用的帐号":"","
  "user":"repl",
  "passwd":"repl4slave"
},

"backup":
{
  "xenon对宣布的IP":"写本地的IP",
  "ssh-host":"172.18.0.10",
  "ssh-user":"root",
  "ssh-passwd":"",
  "basedir":"/usr/local/mysql",
  "从远程备份过来的文件直接放的位置":"指定为datadir",
  "backupdir":"/data/mysql/mysql3306/data",
  "xtrabackup-bindir":"/usr/bin/"
},

"rpc":
{
  "request-timeout":500
},

"log":
{
  "level":"INFO"
}
}
```

# xenon启动

- `nohup ./bin/xenon -c /etc/xenon/xenon.json >./xenon.log 2>&1 &`
- `./bin/xenoncli cluster add  
172.18.0.10:8801,172.18.0.11:8801,172.18.0.12:8801`
- 参考: <https://github.com/zhishutech/radon-ebook/blob/master/xenon/4-xenon-init.md>

# xenon HA

- xenon会成为：mysqld\_safe的守护进程，调用mysqld\_safe启动mysql

```
"admin": "root",  
"passwd": "",  
"host": "127.0.0.1",  
"port": 3306,  
"basedir": "/usr/local/mysql",  
"MySQL启动的配置文件": "",  
"defaults-file": "/data/mysql/mysql3306/my3306.cnf"
```

**/usr/local/mysql/bin/mysqld\_safe --defaults-file=/data/mysql/mysql/mysql3306/my3306.cnf &**

**<https://github.com/radondb/xenon/blob/master/src/raft/leader.go#L332>**



# xenon HA

- 核心原理
  - 基于Raft选举出leader: 会把其它节点change到自己身上, 然后关闭super\_read\_only, read\_only, 调用leader-start-command 绑定vip.
  - 当原来的主节点故障或是crash, xenon会触发一次新的选举, 原始的leader释放vip, 成为follower其它节点提议成为新的Leader, 当成为leader后, 还是走的上面的流程。

# xenon节点重建功能

- 节点重建是使用的xtrabackup实现 调用命令: xenoncli mysql rebuildme 这里调用的配置是xenon.json中backup部分:

```
"backup":  
{  
  "xenon对宣布的IP": "写本地的IP",  
  "ssh-host": "172.18.0.10",  
  "ssh-user": "root",  
  "ssh-passwd": "",  
  "basedir": "/usr/local/mysql",  
  "从远程备份过来的文件直接放的位置": "指定为datadir",  
  "backupdir": "/data/mysql/mysql3306/data",  
  "xtrabackup-bindir": "/usr/bin/"  
},
```

<https://github.com/radondb/xenon/blob/master/src/cli/cmd/mysql.go#L164>

# xenon节点重建功能

节点重建大多会卡在权限这块，一直输出： S11-->start.mysql.end....

临时解决办法： 在11步之前加入：

```
// patch by wubx.  change datadir owner
// 如果运行的用户不是mysql自行修改成需要的用户
{
    datadir := conf.Backup.BackupDir
    cmds := "bash"
    args := []string{
        "-c",
        fmt.Sprintf("chown -R mysql:mysql %s", datadir),
    }

    _, err := common.RunCommand(cmds, args...)
    ErrorOK(err)
    log.Warning("SPre11-->chown owner to mysql[%v]", datadir)
}
```

# 实战演示一下环境

# xenon使用感受

- 好的方面
  - vip控制或是向云系统中的vip中注册比较方便
  - 节点重建也比较方便
- 限制
  - 增强半同步
  - 基于GTID+Row格式复制



# 大纲

- 本次培训目标
- xenon的介绍及特点
- xenon部署及使用
- 对xenon使用上的一些想法

# xenon一些想法

- 支持异步复制，加入mysqlbinlog server支持及数据补偿
- 支持MGR
- 支持注册服务到consul中，需要在xenon中实现守护进程
- 更丰富的Agent功能，RPC接口或是HTTP接口扩展

# QA环节

## 关于「3306π」社区

围绕 **MySQL** 核心技术，将互联网行业中最重要数据化解决方案带到传统行业中；囊括其他开源技术Redis、MongoDB、Hbase、Hadoop、ElasticSearch、Storm、Spark等；分享干货知识，即便是赞助商，也要求如此，拒绝放水。



社区公众号



社区QQ群